VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

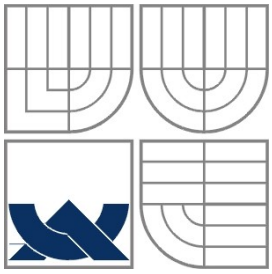
SYSTÉM PRO SPRÁVU MALÉ PODNIKOVÉ SÍTĚ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

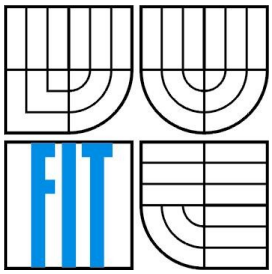
AUTOR PRÁCE
AUTHOR

Bc. JAN HOLEŠINSKÝ

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO SPRÁVU MALÉ PODNIKOVÉ SÍTĚ

MONITORING AND CONTROL SYSTEM FOR SMALL COMPANY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JAN HOLEŠINSKÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Mgr. Roman Trchalík, Ph.D.

BRNO 2014

Abstrakt

Cílem této diplomové práce je navrhnout a implementovat informační systém pro aktivní i pasivní správu malé podnikové sítě s důrazem na co největší automatizovanou správu a nasazení v produkční síti. Systém je vhodně navržen pomocí Nette frameworku a skriptovacího jazyka Perl, který je vhodný na správu počítačových systémů a sítí.

Abstract

The goal of this thesis is to design and to implement the active and passive monitoring and control system for small company with emphasis on the most automated control and deployment in a production network. System is appropriately designed using the Nette framework and scripting language Perl which is suitable for control computer systems and networks.

Klíčová slova

Správa sítě, počítačová síť, směrovač, server, síťové služby, SNMP, Nmap, NetFlow, RRDTool, Perl, Nette framework, MVC, MySQL, Apache, dibi, PHP, AJAX

Keywords

Network control, computer network, router, server, network services, SNMP, Nmap, NetFlow, RRDTool, Perl, Nette framework, MVC, MySQL, Apache, dibi, PHP, AJAX

Citace

Holešinský Jan: Systém pro správu malé podnikové sítě, semestrální projekt, Brno, FIT VUT v Brně, 2014

Systém pro správu malé podnikové sítě

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Mgr. Romana Trchalíka, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Bc. Jan Holešinský
3.6.2014

Poděkování

Děkuji vedoucímu mé diplomové práce Mgr. Romanu Trchalíkovi, Ph.D. za rady, věcné připomínky a nasměrování při řešení. Rovněž svým nejbližším za toleranci při dokončování této práce.

Obsah

Obsah 1
1 Úvod 2
2 Analýza požadavků 3
2.1 Klientské stanice 3
2.2 Podnikové servery 4
2.3 Síťové prvky 5
2.4 Síťová zařízení 5
2.5 Statistiky a sběr dat síťového provozu 6
2.6 Správa systému 6
3 Návrh systému 7
3.1 Návrh topologie testovací sítě 7
3.2 Diagram případů užití 8
3.3 Datový model 9
3.4 Způsob sledování síťových služeb 9
4 Použité technologie 11
4.1 Nette framework 11
4.2 RRDtool 12
4.3 NFDump 12
4.4 Nmap 13
5 Implementace systému 17
5.1 Příprava systému 17
5.2 Přihlášení do systému 18
5.3 Evidence klientských stanic 19
5.4 Monitorování serverů a jejich služeb 22
5.5 Monitorování síťových zařízení 24
5.6 Sledování síťového provozu 25
5.7 Generování statistik 28
6 Instalace 30
7 Závěr 31

1 Úvod

Dnešní význam označení počítačových sítí je mnohem odlišný od sítí před řádově několika desítkami let, kdy základ sítě byl tvořen několika málo klientskými stanicemi a jedním serverem, který ve většině případů představoval, kromě sdíleného úložiště dat, také hlavní bránu pro přístup do celosvětové sítě Internetu. Tento server povětšinou poskytoval také základní tiskové služby, avšak kvalita a datová propustnost, v závislosti na typu média, byla na dnešní poměry velmi nízká. S postupem času a vývojem nových technologií nabyla počítačová síť mnohem komplexnější a flexibilnější vlastnosti, které může svým uživatelům nabídnout. Množství nabízených síťových služeb tedy mnohonásobně vzrostlo a tím i složitost správy při zachování požadované kvality. V dnešní době je téměř nemožné vyčlenit lidské zdroje pro neustálé sledování správné funkčnosti každého síťového zařízení za účelem minimalizace výpadků služeb, popř. snížení jejich poskytované kvality. Díky tomu, že je v dnešní době téměř vše propojeno právě touto sítí, je zde také požadavek udržovat maximální dostupnost základních síťových služeb, jako jsou doménové služby, hlasové služby nebo méně náročné služby jako např. elektronická pošta. Pro menší počítačové sítě existuje spousta základních monitorovacích systémů, ale v převážné většině sleduje jen dostupnost daného zařízení na síti pomocí takzvané síťové odezvy, avšak samotná služba běžící na zařízení může být vzhledem k selhání nebo vytížení systému nedostupná. Proto je potřeba sledovat i síťové služby běžící na konkrétních síťových portech daného serveru a tím zajistit včasné řešení vzniklého problému a eliminovat potencionální výpadek části sítě nebo dalších závislých služeb.

Rozhodl jsem se tedy v této práci zaměřit právě na aktivní a pasivní správu sítě včetně již zmíněných serverových síťových služeb doplněnou o sběr dat ze síťového provozu za účelem pozdější analýzy využití sítě a vhodného budoucího rozšíření stávající infrastruktury.

2 Analýza požadavků

Předem je nutno říci, že síť je tvořena třemi základními skupinami zařízení, díky kterým jsou uživatelé schopni využívat nabízené síťové služby. První skupina zařízení zahrnuje všechny tzv. prvky sloužící k propojování sítí a připojování koncových klientských stanic včetně serverů. Jedná se především o směrovače a rozbočovače. Další skupinou zařízení jsou zařízení, která přímo neslouží pro fyzické propojení sítí, ale jsou důležité právě pro poskytování síťových služeb. Zmínil bych zde např. doménové nebo poštovní servery, síťové tiskárny, faxy, IP kamery, IP telefony atd. Třetí skupinu tvoří koncové uživatelské stanice připojené drátově či bezdrátově, které až na pár výjimek neposkytují žádné síťové služby. V dnešní době se zdá být tato skupina nejrozmanitější, jelikož lze využívat síťové služby ne jen ze stolních počítačů nebo notebooků, ale i chytrých telefonů, tabletů, televizních přijímačů nebo také náramkových hodinek.

Aby bylo možné uceleně navrhnout vhodný systém pro správu sítě, byly definovány následující požadavky.

2.1 Klientské stanice

Jak již bylo zmíněno, tato skupina je tvořena uživatelskými zařízeními, které není třeba v počítačové síti centrálním systémem aktivně sledovat nebo jakkoliv spravovat. Vhodné je však každou stanici evidovat a případně i sledovat její provoz na síti.

Na základě této analýzy by měl navrhovaný systém evidovat všechny podnikové i nepodnikové klientské stanice včetně připojených mobilních zařízení, u nichž by měla být uvedena využívaná IP adresa, název stanice, popř. název běžícího operačního systému. Stanice by měla být rovněž specifikována kategorií způsobu připojení, zda využívá pevné připojení k síti nebo bezdrátové připojení. U bezdrátového připojení je vhodné evidovat název užívaného přístupového bodu. Rovněž datum o posledních evidovaných aktivitách dané stanice bude mít užitečný informativní charakter.

Pro účely sledování využívání sítě budou generovány statistiky v podobě grafů pro předem zvolené síťové služby a také celkový přenos dat pro každou klientskou stanici.

Způsob zadávání nových stanice bude umožněno přímo správcem systému nebo prohledáním sítě samotným dohledovým systémem, a to rozpoznáváním všech nových

dostupných stanic v síti nebo konkrétních stanic dle specifikované IP adresy, popř. síťového rozsahu.

2.2 Podnikové servery

Oproti klientským stanicím, podnikové servery budou vyžadovat nejen pasivní, ale také aktivní sledování poskytovaných síťových služeb. Zde již požadavky na dostupnost serverových služeb klientům vzrůstá, a proto si nevystačíme jen se sledováním síťového provozu, ani se sledováním odezvy od serveru jakožto zařízení připojeného do počítačové sítě.

Mimo základní požadavky na evidenci serverů, jako je název serveru, IP adresy, operační systém, popř. doba běhu serveru, bude nutné evidovat i spuštěné služby (DNS, SMTP, POP3, HTTP atd.) poskytované klientským stanicím, a ne jen jim, v rámci počítačové sítě. Stav evidovaných běžících služeb bude zjišťován aktivním sledováním a v případě výpadku služeb, resp. jejich nedostupnosti, bude správci sítě zaslána upozorňující zpráva. Jako pasivní sledování serverů bude možné zobrazovat záznamy síťového provozu vztahující se ke konkrétnému zařízení, popř. zvolené síťové službě. Tato data budou rovněž vhodná při řešení různých síťových problémů, jako jsou DoS útoky, které mají za účel znepřístupnit poskytovanou službu.

Servery bude také možné aktivně spravovat prostřednictvím SSH přístupu, kdy budou zpřístupněny pro běžící serverové služby základní odkazy spouštějící terminálové příkazy dle definovaného operačního systému. Dále bude možné zobrazovat aktuální statistiky síťových rozhraní pro případ ztrátovosti přijímaných rámců vlivem vadné kabeláže nebo dokonce fyzickým síťovým rozhraním. Pro každý server bude taktéž generován graf znázorňující statistiku přenosu dat.

Způsob vytváření nových záznamů, tj. vkládání nových serverů do evidence bude možné ručním zadáním nebo vyhledáváním a rozpoznáváním nových serverů v celé síti nebo ve správcem definovaném rozsahu sítě, popř. zvolením konkrétní IP adresy. Systém se tedy pokusí detekovat již běžící služby, které poté navrhne při ukládání do seznamu.

2.3 Síťové prvky

Nedílnou součástí každé sítě jsou právě síťové prvky jako např. směrovače, rozbočovače a bezdrátové přístupové body. Na těchto prvcích je postavena celá síť a proto je vhodné sledovat i tyto zařízení aktivním způsobem.

Základním požadavkem je evidence názvu síťového prvku, IP adresy, doba běhu zařízení, značka výrobce a jeho modelové označení. V rámci pasivního sledování bude možné sledovat toky dat procházející těmito síťovými prvky. Vhodným způsobem, čili aktivním sledováním, bude také možné zobrazit aktuální statistiky síťových rozhraní. V případě bezdrátových přístupových bodů bude možné sledovat počet aktivně připojených uživatelů s odkazem na jejich stanice. V neposlední řadě systém umožní generování grafů se statistikami vytížení procesorů a systémové paměti. Graficky znázorněné budou také statistiky o celkovém obousměrném toku dat na odchozích síťových rozhraních zařízení.

2.4 Síťová zařízení

Další skupinou jsou uživatelská síťová zařízení. Oproti klientským stanicím jsou tyto zařízení sdíleny všem uživatelům prostřednictvím počítačové sítě, avšak svou jednoúčelovostí se nejedná o všestranné počítačové stanice. Svým způsobem se tedy jedná o síťové servery nabízející obvykle jen jednu službu. Názorným příkladem je síťová tiskárna, která běžně nabízí tiskové služby, popř. skenování v případě multifunkčního zařízení. Stejně je tomu tak i u tzv. IP telefonů nebo hlasové brány VoIP, která poskytuje pouze hlasové služby prostřednictvím sítě Internet. Jelikož jsou tyto zařízení ve většině případů integrovaná, není nám umožněno běžně přistupovat do systému, avšak jsou povoleny omezené funkcionality za účelem jejich správy a využití. Tato zařízení tedy budeme obvykle sledovat jen pomocí dotazů na odezvu od síťového rozhraní sledovaného zařízení nebo pomocí SNMP dotazů, je-li to možné.

Záznamy o síťových zařízeních budou běžně evidovat název zařízení, IP adresu, výrobce a modelové označení. Rozlišovat se také bude typ zařízení podle nabízených funkcí, tj. nabízí-li tiskové, skenovací a kopírovací služby, bude toto zařízení označeno jako multifunkční tiskárna s dodatkem, jaké funkce jsou nabízeny. Rovněž bude vhodné pasivním způsobem sledovat síťový provoz zařízení, jako u ostatních síťových zařízení.

2.5 Statistiky a sběr dat síťového provozu

Pro kvalitní správu každé počítačové sítě jsou velmi užitečné právě záznamy o síťovém provozu, a to ne jen o provozu z klientských stanic, ale také z podnikových serverů, na které mohou být vedeny již zmíněné různé typy útoků nebo pouze velký počet dotazů, které není server schopen v reálném čase zpracovávat vlivem výkonově slabé výpočetní platformy.

Na základě získaných dat ze síťového provozu budou průběžně generovány grafické statistiky využívaných standardních síťových služeb. Rovněž bude možné prohlížet záznamy o síťovém provozu všech připojených zařízení dle zvolených zdrojových nebo cílových IP adres, popř. blíže specifikovaných parametrů pro filtrování hledaného provozu.

2.6 Správa systému

V poslední sekci bude přednastaveno veškeré nastavení běhu systému, které bude možné přizpůsobit konkrétním požadavkům správce sítě. Základní součástí nastavení budou tzv. číselníky využívané pro bližší specifikaci evidovaných podnikových zařízení. Tyto základní číselníky bude možné upravovat, popř. přidávat nové. Bude zde definováno, kam zaslat upozorňující zprávu při zjištěném výpadku některé síťové služby nebo nedostupnosti síťového zařízení. Nebude chybět ani nastavení odchozího poštovního serveru, prostřednictvím kterého budou informační zprávy zasílány.

Další důležitou součástí bude správa SNMP seznamu, kde budou evidovány tzv. identifikátory objektů OID sloužící pro získání libovolných údajů stejnojmenným standardizovaným protokolem SNMP z podporovaných zařízení. U těchto identifikátorů bude možné zvolit, pro jaký typ zařízení budou informace získávány.

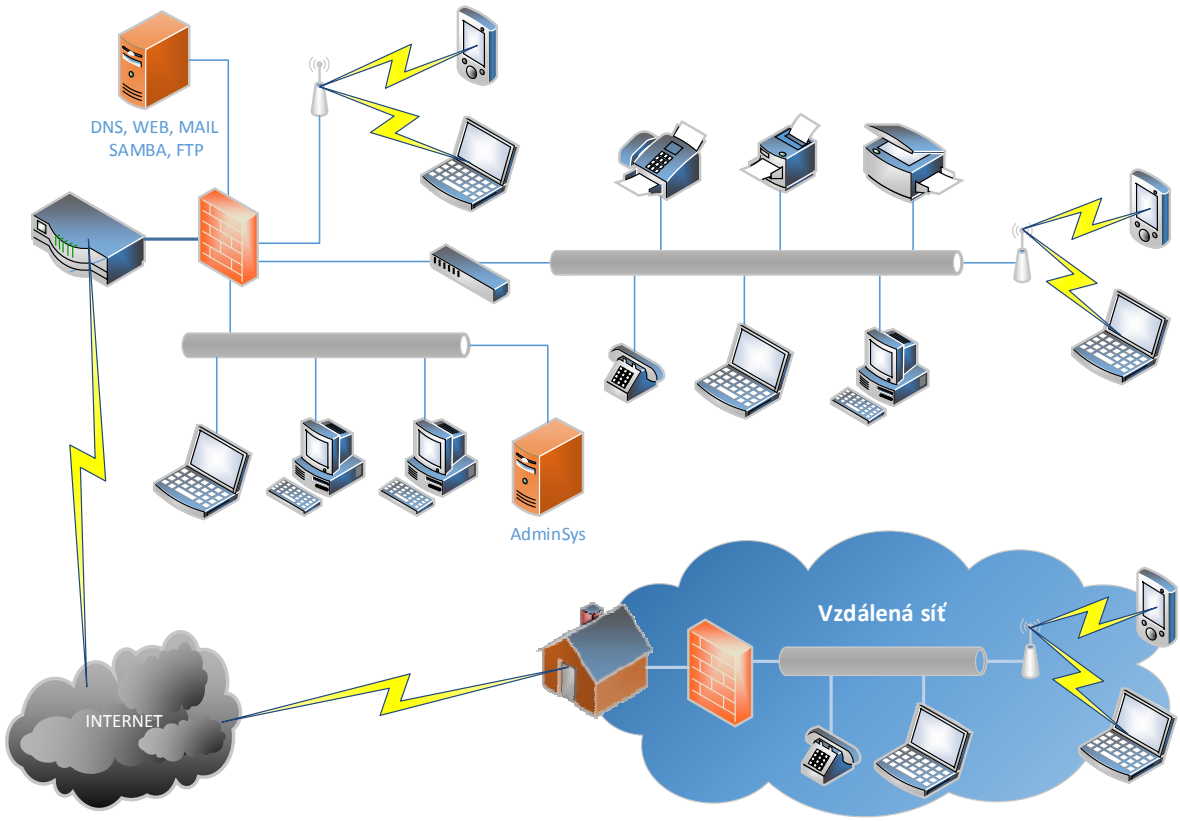
Součástí správy bude také sledování stavu naplánovaných skriptů sloužící pro shromažďování informací o provozu na síti a pro sledování dostupnosti síťových služeb. Tyto naplánované skripty bude možné aktivovat i deaktivovat přímo ze systému.

3 Návrh systému

Jeden z požadavků na navrhovaný systém je co nejvíce automatizovat správu. Systém by tak měl od uživatele požadovat pouze základní informace nezbytné pro identifikaci daného zařízení a volitelné informace na základě již známých dat dohledat prostřednictvím vlastních prostředků, jako např. aktivní analýzou zařízení pomoci SNMP protokolu nebo prohledáváním celé sítě. Pro účel návrhu systému a jeho testování při inkrementální implementaci budeme uvažovat následující navrženou topologii testovací sítě.

3.1 Návrh topologie testovací sítě

Testovací síť znázorněná na obrázku 3.1 je navržena s ohledem na testování správy všech požadovaných typů zařízení, tj. podnikový server nabízející základní síťové služby, tj. doménové služby, příchozí a odchozí elektronická pošta včetně webových služeb.

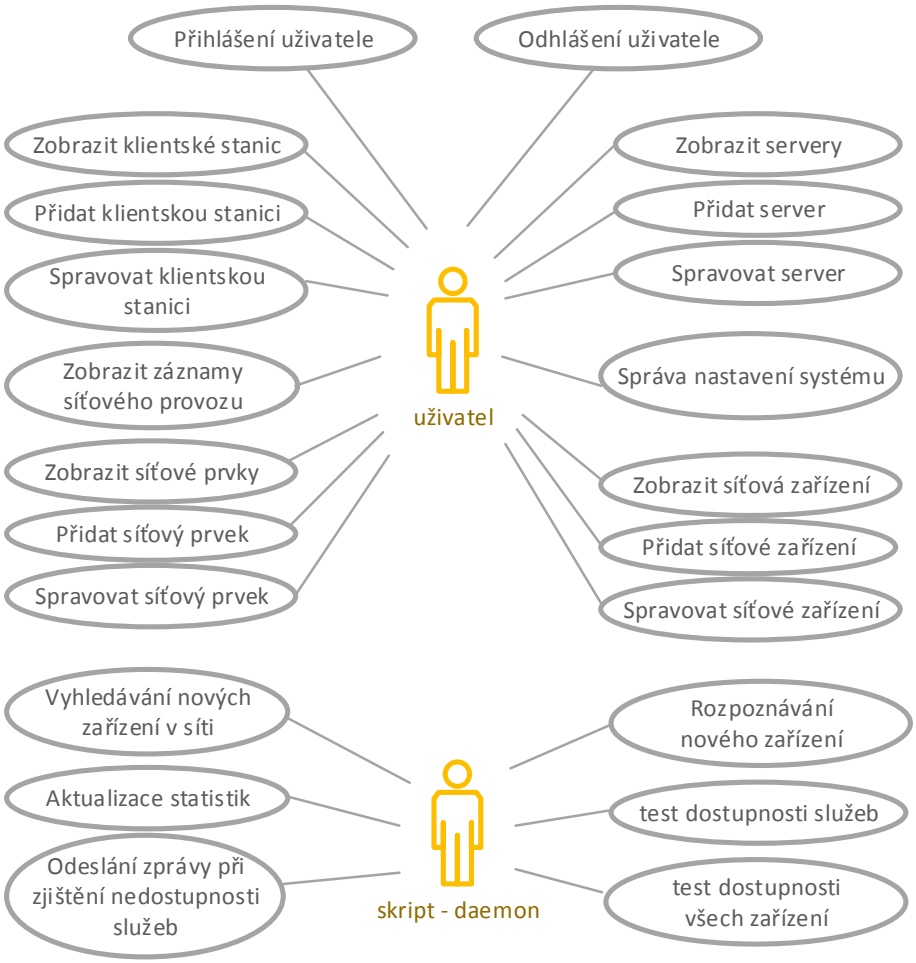


Obrázek 3.1: Topologie testovací sítě

Dále síťová zařízení, jako jsou tiskárny, fax a Internetová hlasová brána VoIP. Ze síťových prvků jsou zastoupeny směrovače, rozbočovače a bezdrátové přístupové body. Některé z nich však umožňují správu pouze prostřednictvím webového rozhraní. V neposlední řadě bych také zmínil klientské koncové stanice, jako jsou stolní počítače, notebooky, bezdrátové chytré telefony atd. Pro pozdější testování je topologie rozšířena o tzv. vzdálenou pobočkovou síť, která bude s místní sítí propojena zabezpečeným virtuálním tunelem VPN, který nám umožní sledovat vzdálenou síť, jakoby byla fyzickou součástí místní sítě.

3.2 Diagram případů užití

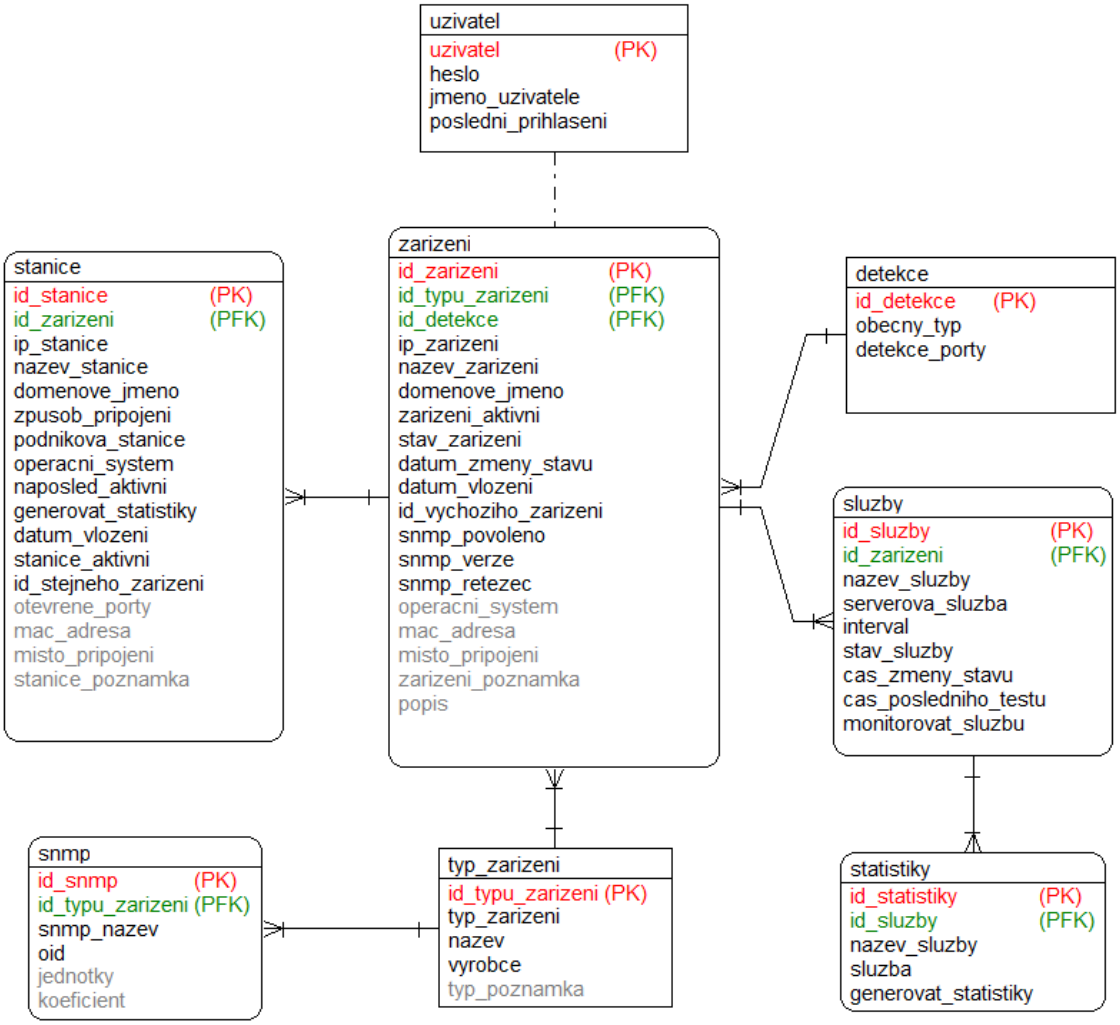
Pro návrh systému byly nejdříve definovány uživatelské role a jejich případy užití, ze kterých se dále definují entity tvořící základ konceptuálního modelu. Případy užití byly navrženy tak, aby uživatel systému dostával automaticky získané informace a ty dle potřeby jakkoliv upravoval.



Obrázek 3.2: Diagram případů užití

3.3 Datový model

Na základě předchozího diagramu byly zvoleny entity, které tvoří základ datového modelu pro návrh databázové struktury (viz obrázek 3.3). Vzhledem k efektivnosti a udržitelnosti co nejmenší velikosti databáze bude v systému využito více databázových typů. Mimo standardní databázi bude využíváno tzv. cyklické RRD databáze s konstantní velikostí dat. Více informací o této technologii bude popsáno v další kapitole.



Obrázek 3.3: Databázový diagram

3.4 Způsob sledování síťových služeb

Jak již bylo zmíněno v úvodu, budeme ověřovat dostupnost síťových služeb odesíláním testovacích zpráv na dané servery a na základě přijaté odpovědi analyzovat, zda služba

pracuje správně nebo nikoliv. Způsoby testování těchto služeb se ovšem liší v závislosti na používaném protokolu.

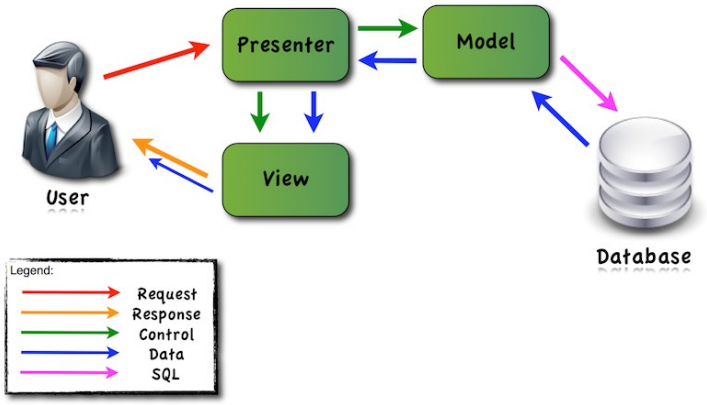
Běžné textové protokoly využívající TCP spojení [1] je tak možné jednoduše otestovat pouhým zasláním dotazu na konkrétní číslo portu. V případě, že nám server odpoví, můžeme říci, že služba běží. Pokud však detailněji zanalyzujeme přijatou odpověď, můžeme také zjistit, zda služba běží správně nebo vykazuje nestandardní chování. V případě UDP spojení, které využívá např. doménová služba (port UDP/53), si s běžnými testy nevystačíme. Pro tento případ bude nutné využít některé ze standardních testovacích funkcí (např. nslookup) nebo využití knihovny skriptovacího jazyka Perl. Některé služby však využívají šifrování SSL. V tomto případě se bez pomocné knihovny neobejdeme. Naštěstí i tuto situaci vhodně řeší knihovna Perlu. Systém bude standardně testovat všechny dostupné porty pro jejich odpověď, avšak správnost běhu testovaných služeb bude ověřovat jen pro nejpoužívanější z nich. V budoucnu však bude možné tuto funkcionalitu rozšířit přidáním dalších testovacích skriptů pro konkrétní služby.

4 Použité technologie

Aby byl systém dostupný odkudkoliv, poběží jako webová aplikace za pomocí moderního PHP Nette Frameworku, který je v současné době dostupný ve verzi 2.2. Vzhledem k tomu, že se tato poslední verze od předchozí značně liší a její dokumentační část ještě není úplně aktualizována, budu v další části textu uvažovat právě předchozí stabilní verzi, tj. verzi 2.1.2.

4.1 Nette framework

Hlavní předností Nette frameworku [2] je dokonalé zabezpečení využívající revoluční technologii, která eliminuje výskyt bezpečnostních děr a jejich zneužití, jako je např. tzv. Cross-Site Scripting (zkráceně XSS), Session hijacking a další útoky. Velkou výhodou je také zabudovaný ladící nástroj, který v průběhu implementace odhalí chyby a velmi srozumitelným způsobem nám vypíše, kde se přesně nachází. Základem Nette Frameworku je architektura zvaná MVP [3], neboli Model-View-Presenter. Původem se jedná o softwarovou architekturu rozdělující aplikaci právě do tří základních vrstev (viz obrázek 4.1). První vrstva, tj. Model, tvoří datový a hlavně funkční základ aplikace. Zde jsou uvedeny veškeré operace nad databázovými či jinými daty. S touto vrstvou komunikuje jen tzv. Presenter, který tvoří další vrstvu této architektury. Presenter je objekt přijímající požadavky od uživatele a předává získaná data z modelu prostřednictvím pohledu (vrstva View) zpět uživateli. Poslední vrstva View pouze předává zpracovaná data zpět uživateli.



Obrázek 4.1: Architektura MVP [3]

Datová část bude implementována pomocí dvou databází, kde systémová data budou ukládána pomocí MySQL a statistiky ze sítě v cyklické round-robin databázi (dále jen RRD). Důvod použití RRD je možnost ukládání dat do souborů s konstantní velikostí a hlavně rychlím přístupem k datům.Komunikace se systémovou databázi MySQL bude rovněž snadná díky použití databázové vrstvy Dibi [9], která společně s Nette Frameworkem tvoří silný vývojový nástroj. Dibi umožňuje tzv. plynulé dotazování, které nám umožňuje snadno a přehledně skládat libovolný databázový SQL dotaz. Příkladem tohoto zápisu je vyhledání všech záznamu ze dvou spojených tabulek:

```
public function findAllDevices() {
     return $this->database->select('*')->from('zarizeni')
                    ->where("obecny_typ = 'zarizeni'")
                    ->leftJoin('typ_zarizeni')->using('(id_typu_zarizeni)')
                    ->orderBy('ip_zarizeni');
}
```

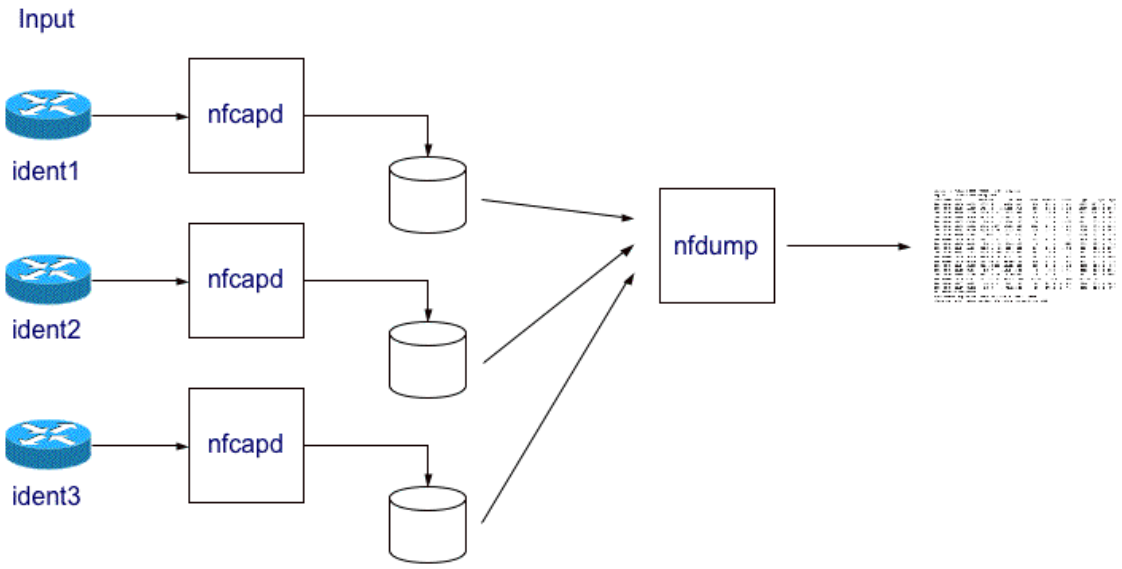
4.2 RRDtool

Round-robin database tool, neboli RRDtool [8] je volně dostupný nástroj šířený pod licencí GNU (General Public Licence), který umožňuje spravovat časově závislá data a ukládat je do cyklické databáze, která funguje jako fronta FIFO. Znamená to tedy, že data se zapisují do databáze do té doby, než se celá zaplní. Poté se pro vložení dalšího záznamu odstraní první vložený záznam a nový záznam se vloží na konec fronty, tj. za poslední záznam. Tím je dosaženo toho, že tato databáze má konstantní velikost. Nástroj je podporován širokou škálou skriptovacích jazyků, jako je Perl, Python nebo Ruby. V této práci jsem však zvolil skriptovací jazyk Perl. Využití tohoto nástroje bude především pro ukládání získaných statistik, ale také pro generování průběžných grafů.

4.3 NFDump

Další volně dostupným nástrojem šířený pod licencí BSD jsou NFDump nástroje [10]. Tento nástroj obsahuje jednotlivé aplikace pro sběr a správu dat síťového provozu pomocí protokolu NetFlow ve verzích 5, 7 a 9. NetFlow protokol se sestává ze dvou hlavních částí,

sondy a collectoru (sběrač dat). V případě NFDump nástrojů je sběrač dat pojmenován jako `nfcapd`, který se spouští v režimu deamona. Síťový provoz je poté transparentně ukládána do jednotlivých souborů podle data a času přijetí. Pro čtení uložených dat využijeme `nfdump`. Tento nástroj umožňuje rychlé vyhledávání i za použití filtračních kritérií. Dokáže nám vyhledat určitý předem specifikovaný síťový provoz i jejich počty s průměrnými přenosovými rychlostmi. Tímto způsobem jsem schopni vyhledat např. pokusy o cílený útok na některé síťové zařízení, resp. jeho služby. Schéma shromažďování síťového provozu pomocí nástroje NFDump je na obrázku 4.3.



Obrázek 4.3: Proces zpracování síťového provozu [10]

4.4 Nmap

Dále budu vycházet z [4]. Jedná se o jeden z nejvyužívanějších nástrojů právě pro skenování sítí a prověřování jejich zabezpečení. Je to velmi pokročilý nástroj, který dokáže kromě skenování sítě také skenování portů, vyhledávat běžící počítače a detekovat jejich služby. Dokonce zjistí i verze běžících služeb a verze operačního systému. Základní použití nástroje Nmap je následující:

```
nmap [parametry] [cílová adresa]
```

Jednou z důležitých voleb je rychlost skenování. To však záleží na typu sítě a způsobů zabezpečení. V případě, že se na síti vyskytuje tzv. intrusive detection system (dále jen IDS)

pro detekci útoků, je vhodné použít právě pomalejší skenování, tj. parametry `T1` až `T3`. Chceme-li nová zařízení vyhledat rychleji a prohledávání provádíme na síti LAN, můžeme zvolit agresivnější režim s parametrem `T aggressive` neboli `T4`. Ve výjimečných případech lze použít i nejrychlejší způsob s parametrem `T5`, ten ovšem doporučuji jen v laboratorním prostředí.

Příklad analýzy zařízení v testovací síti:

```
# nmap -T4 192.168.10.8
Starting Nmap 5.51 ( http://nmap.org ) at 2014-05-01 13:04 CEST
Nmap scan report for wifi.jahol.local (192.168.10.8)
Host is up (0.00046s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
2000/tcp open  cisco-sccp
8291/tcp open  unknown
MAC Address: 4C:5E:0C:2B:F0:CA (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.68 seconds
```

Další užitečný parametr je `-p`, který umožní skenovat určité čísla portů:

```
# nmap -T4 -p 22,136-139,443 192.168.10.2
Starting Nmap 5.51 ( http://nmap.org ) at 2014-05-01 13:19 CEST
Nmap scan report for fileserver.jahol.local (192.168.10.2)
Host is up (0.00062s latency).
PORT    STATE  SERVICE
22/tcp  open   ssh
136/tcp closed profile
137/tcp closed netbios-ns
138/tcp closed netbios-dgm
139/tcp open   netbios-ssn
443/tcp open   https
MAC Address: 00:11:32:18:09:6D (Synology Incorporated)

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

V neposlední řadě bych se zmínil o makru `-A`, které využívá mnoho typických funkcí pro detekci verzí služeb, operačního systému, traceroutu atd. Výstupem takového příkazu je následující příklad:

```
# nmap -A 192.168.10.10
Starting Nmap 5.51 ( http://nmap.org ) at 2014-05-01 13:25 CEST
Nmap scan report for vbox-srv.jahol.local (192.168.10.10)
Host is up (0.00053s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn
445/tcp  open  netbios-ssn
3389/tcp open  microsoft-rdp Microsoft Terminal Service
MAC Address: 00:14:85:F7:DB:5E (Giga-Byte)
Warning: OSScan results may be unreliable because we could not find at
least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows 2008|7|Vista
OS details: Microsoft Windows Server 2008, Microsoft Windows 7
Professional, Microsoft Windows Vista SP0 or SP1, Server 2008 SP1, or
Windows 7
Network Distance: 1 hop
Service Info: OS: Windows
```

Budeme-li chtít rychle prohledat lokální síť bez hlubší analýzy jednotlivých zařízení, můžeme použít parametr `sP` spolu s již zmíněným parametrem `T4` a získáme rychlý přehled zařízení.

```
# nmap -sP -T4 192.168.10.0/24
Starting Nmap 5.51 ( http://nmap.org ) at 2014-05-01 13:58 CEST
Nmap scan report for gw-router.jahol.local (192.168.10.1)
Host is up (0.00064s latency).
MAC Address: 00:0C:42:A9:33:C5 (Routerboard.com)
Nmap scan report for mail.jahol.local (192.168.10.2)
Host is up (0.00036s latency).
MAC Address: 00:11:32:18:09:6D (Synology Incorporated)
Nmap scan report for tisk.jahol.local (192.168.10.3)
Host is up (0.00048s latency).
```

```
MAC Address: 00:21:5A:E5:47:57 (Hewlett Packard)
Nmap scan report for dns.jahol.local (192.168.10.4)
Host is up (0.00030s latency).
MAC Address: 00:0C:42:A9:33:C5 (Routerboard.com)
Nmap scan report for voip.jahol.local (192.168.10.5)
Host is up (0.0029s latency).
MAC Address: 7C:2F:80:2D:50:32 (Gigaset Communications GmbH)
Nmap scan report for fax.jahol.local (192.168.10.6)
Host is up (0.00075s latency).
MAC Address: B0:FA:EB:31:98:10 (Unknown)
Nmap scan report for wifi.jahol.local (192.168.10.8)
Host is up (0.00052s latency).
MAC Address: 4C:5E:0C:2B:F0:CA (Unknown)
Nmap scan report for adminsys.jahol.local (192.168.10.9)
Host is up.
Nmap scan report for vbox-srv.jahol.local (192.168.10.10)
Host is up (0.00026s latency).
MAC Address: 00:14:85:F7:DB:5E (Giga-Byte)
Nmap scan report for ntb-jahol.jahol.local (192.168.10.15)
Host is up (0.00042s latency).
MAC Address: 20:CF:30:54:0C:B7 (Asustek Computer)
Nmap done: 256 IP addresses (11 hosts up) scanned in 2.25 seconds
```

5 Implementace systému

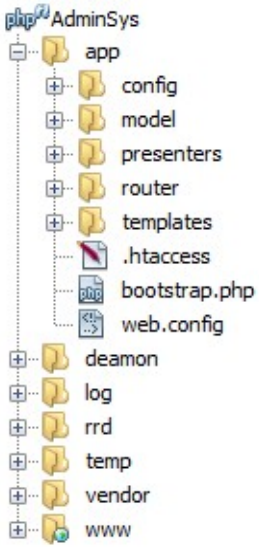
Pro snadnější implementaci a využití širokého spektra dostupných systémových nástrojů poběží navrhovaný systém na linuxové platformě. Konkrétně se jedná o distribuci CentOS 6.5 vycházející z Red Hat Enterprise Linux distribuce určené pro firemní využití. Server bude v počítačové síti umístěn tak, aby byl co nejblíže hraničnímu směrovači a bylo tak možné zajistit informovanost správce při jakémkoliv výpadku služeb nebo nedostupnosti zařízení v lokální síti (viz obrázek 3.1). Toto umístění také hraje roli při zjišťování trasy výpadku síťových prvků. Pokud tedy sledovací systém nemá odezvu od nejbližšího síťového prvku, nebude mít s velkou pravděpodobností odpověď ani ze vzdálenějšího prvku. Tímto způsobem tedy bude možné snadněji odhalit nefunkční zařízení.

5.1 Příprava systému

Jak již bylo zmíněno v předchozí části, zvolený operační systém je aktuální linuxová distribuce CentOS (v současnosti verze 6.5). Pro snadnější nasazení systému do reálné sítě a případnou migraci na jiné zařízení jsem systém nainstaloval jako virtualizovaný server ve VirtualBoxu využívající již existující klientskou stanici nebo server v produkční síti. Lze tak využít část volných systémových prostředků existujícího serveru, který je téměř nevyužitý. Pro instalaci jsem zvolil již připravenou minimalizovanou verzi distribuce [11], do které jsem doinstaloval potřebné služby jako je webový server Apache, databázová platforma MySQL a RRD, skriptovací jazyky PHP a PERL, SNMP pro správu zařízení, NFdump jako sběrač NetFlow dat ze síťových zařízení a v neposlední řadě Nmap sloužící pro prohledávání a analýzu počítačové sítě. Z důvodu minimalizace výkonového zatížení běžící webové aplikace jsem k danému serveru připojil druhý virtualizovaný disk sloužící čistě pro cyklické uchovávání dat ze síťového provozu. Tento virtuální disk v podobě jednoho souboru je vhodné uložit na nesystémový fyzický disk, čímž docílíme rychlejší odezvy systému i při častém ukládání, popř. čtení, záznamů z datového úložiště. Cesta k tomuto disku je v systému nastavena jako `/mnt/storage`.

Adresářová struktura celého systému je rozdělena několika hlavních částí (viz obrázek 5.1). Aplikační část je uložena ve složce App. Zde jsou rozděleny zdrojové kódy podle architektury Nette frameworku, tzn. složka `model` obsahuje funkce pro získávání dat z databází a jiných zdrojů. Složka `presenters` obsahuje funkce náležící presenterům

jednotlivých webových sekcí a složka `templates` obsahuje jejich pohledy, tj. připravené HTML šablony pro zobrazení získaných dat. Zbylé složky v aplikační části slouží jako nastavení připojení k databázi a inicializaci závislých knihovních funkcí. Další částí základní adresářové struktury je webová část uložená ve složce `www`. Tato složka zároveň slouží jako kořenový adresář webového serveru. Tím zamezíme přímý přístup k ostatním aplikačním složkám a jejich souborům. Webová složka také obsahuje podsložku `graphs`, kde jsou průběžně ukládány generované grafické statistiky síťového provozu. Využívané knihovní funkce jsou uloženy ve složce `vendor`. Zde je uložena distribuce Nette včetně databázové vrstvy Dibi. Složka `log` a `temp` slouží čistě pro logování a odkládání dat administračního systému. Další důležitou složkou je `rrd`, kde jsou uloženy databázové RRD soubory včetně Perl skriptů, které s těmito databázemi pracují a aktualizují jejich data. Ve složce deamon najdeme skripty, které slouží pro získávání síťových dat a práci s administračním systémem.



Obrázek 5.1: Adresářová struktura administračního systému

5.2 Přihlášení do systému

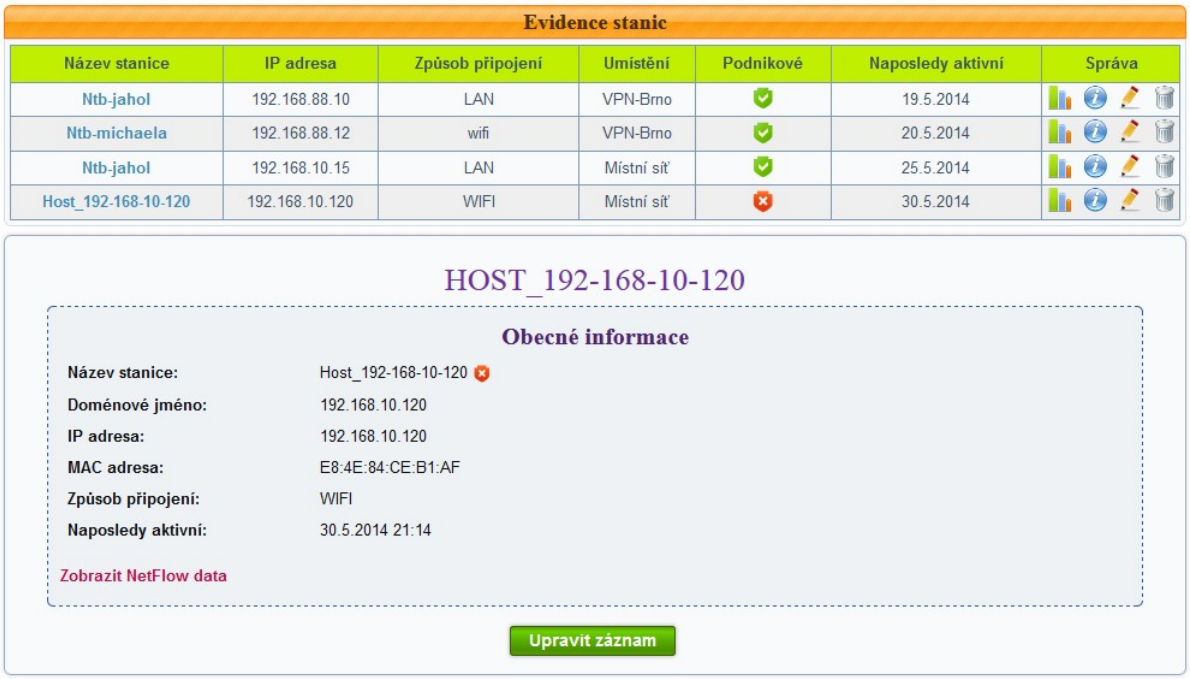
Vzhledem k tomu, že se v systému ukládají důležité informace o celé síti včetně historie síťového provozu, bylo vhodné zabezpečit webovou aplikaci pouze pro jednoho správce. Tato funkcionalita je zajištěna pomocí objektové třídy SignPresenter, ve které nám komponenta SignInForm umožní zobrazit přihlašovací formulář. Je zde možnost zvolit trvalé přihlášení nebo dočasné s omezením na 20 minut nečinnosti, popř. po zavření prohlížeče. V případě, že dojde k odhlášení v průběhu práce, avšak po překročení doby nečinnosti,

systém si uloží naposledy otevřenou stránku a přesměruje uživatele na přihlašovací formulář. Pokud uživatel opět zadá správné přihlašovací údaje, bude díky uloženému poslednímu webovému požadavku (funkce `storeRequest`) přesměrován zpět na původní stránku.

```
protected function startup() {
    if (!$this->user->isLoggedIn()) {
      if ($this->user
            ->getLogoutReason()=== Nette\Security\User::INACTIVITY) {
            $this->flashMessage('Byl jste automaticky odhlášen...');
      }
      $this->redirect('Sign:in',
                      array('backlink' => $this->storeRequest() )
                     );
    }
    parent::startup();
}
```

5.3 Evidence klientských stanic

Všechny stanice, ať již podnikové či nepodnikové, jsou uloženy v databázové tabulce s názvem `stanice`. Tato tabulka v současnosti eviduje pouze IPv4, tj. IPv6 není v tomto systému podporována. Pro zajištění větší automatizace při přidávání nových síťových zařízení do evidence systému jsem se rozhodl využít běžně dostupný nástroj popisovaný v kapitole 4.4 zvaný Nmap. Administrační systém (dále jen AdminSys) využívá výše popsaných funkcionalit právě pro prohledávání sítě na pozadí, kde nová síťová zařízení dokáže jednoduše vyhledat a poté vložit do databáze jako nepodnikové klientské stanice. Pokud dané zařízení se stejnou IP adresou v databázi již existuje, provede se jen časová aktualizace posledního výskytu zařízení v síti. V případě, že se jedná o nové zařízení, záznam se vloží do databáze. Příklad nalezeného síťového zařízení je v grafické podobě na obrázku 5.3.1. Pokud zjistíme, že nově nalezené zařízení je podniková koncová stanice, můžeme ji jednoduše upravit a přidat mezi podnikové.

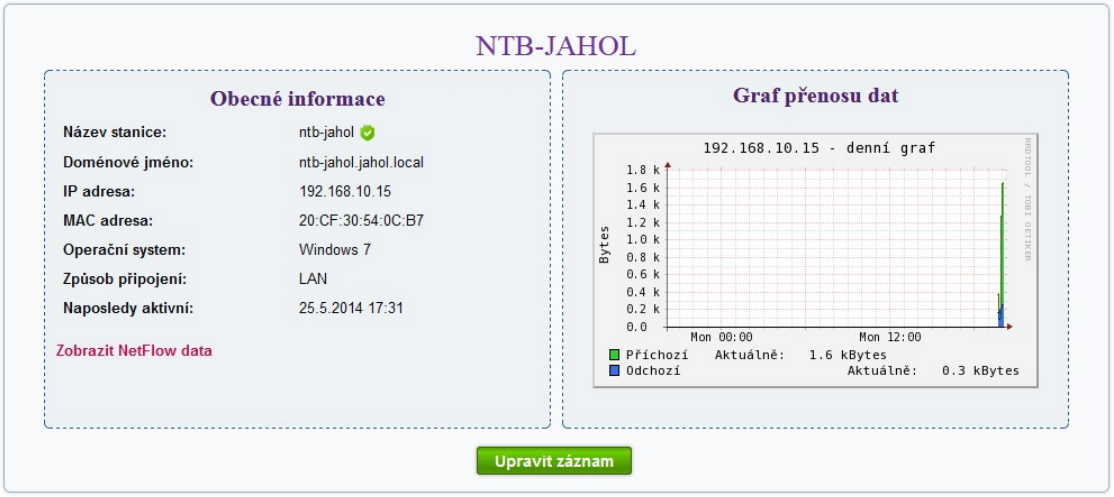


Obrázek 5.3.1: Nově nalezená klientská stanice

V případě, že požadovaná klientská stanice nebyla automaticky vyhledána, je zde také možnost přidat stanici ručně. K tomu nám postačí zadat alespoň IP adresu, popř. další údaje. Zbylé chybějící informace se pokusí systém získat přímo ze sítě. Pro zjištění MAC adresy jsou zde dva způsoby, metoda `getMacByCmd($ip_host)` a `getMacBySnmp($ip_host)`. V první metodě AdminSys odešle echo-request pomocí příkazu `ping`, čímž se odešle do sítě ARP dotaz na MAC adresu přiřazenou námi požadované IP adrese. V případě, že nám zařízení odpoví, AdminSys si získanou MAC adresu přečte v jeho ARP tabulce. Druhý metoda je založena na SNMP dotazování. AdminSys si nejdříve zjistí ze své konfigurace výchozí bránu sítě (IP adresu směrovače) a poté se ji vyhledá ve své databázi síťových prvků. Pokud směrovač nalezne a zjistí, že zařízení podporuje SNMP, odešle směrovači dotaz na jeho ARP tabulku a pokusí se najít hledaný záznam. Zápis druhé metody je zobrazen níže.

```
public function getMacBySnmp($ip_host) {
  $mac_addr = "";
  $default_router = exec(
            "ip route show default | grep default | awk {'print \$3'}"
          );

  if( (int)$this->database
                ->select('count(*)')
                ->from('zarizeni')
                ->where('ip_zarizeni = %s',$default_router)
                ->fetchSingle() ) {

  $router = $this->database
     ->select('`ip_zarizeni`,`snmp_povoleno`,`snmp_retezec`,`snmp_verze`')
     ->from('zarizeni')
     ->where('ip_zarizeni = %s', $default_router)
     ->fetch();

  if($router->snmp_povoleno > 0) {
         snmp_set_quick_print(1);
        $arp_tab = snmprealwalk($default_router,
                                $router->snmp_retezec,
                                $this->snmp_req[5] );

      foreach ($arp_tab as $ip => $mac) {
             $arp_mac = explode('.', $ip, 3);
             if($arp_mac[2] == $ip_host) {
                return $mac;
}}}}
```

U všech evidovaných stanic je možné sledovat historii síťového provozu pro zvolenou IP adresu dle zvoleného období a dalších kritérií. U stanice je tato volba označena jako `Zobrazit NetFlow data`. Záznamy lze vyhledávat i zpětně, tj. přidáme-li klientskou stanici v nějaké době, ale tato stanice již předtím komunikovala se sítí, budou i tyto data dohledány. Více o sledování síťového provozu bude zmíněno v kapitole 5.6. U každého klienta je také možné zapnout generování statistik přenosu dat v grafické podobě. Tato možnost však vychází z aktuálně získávaných hodnot ze síťového provozu (viz obrázek 5.3.2).



Obrázek 5.3.2: Podniková stanice s grafem přenosu dat

5.4 Monitorování serverů a jejich služeb

Jak již bylo zmíněno v předchozí části, všechna zařízení jsou automaticky vyhledávána v určitém intervalu na pozadí a vložena do databáze. Budeme-li chtít přidat do evidence nové serverové zařízení, máme dva možné způsoby.

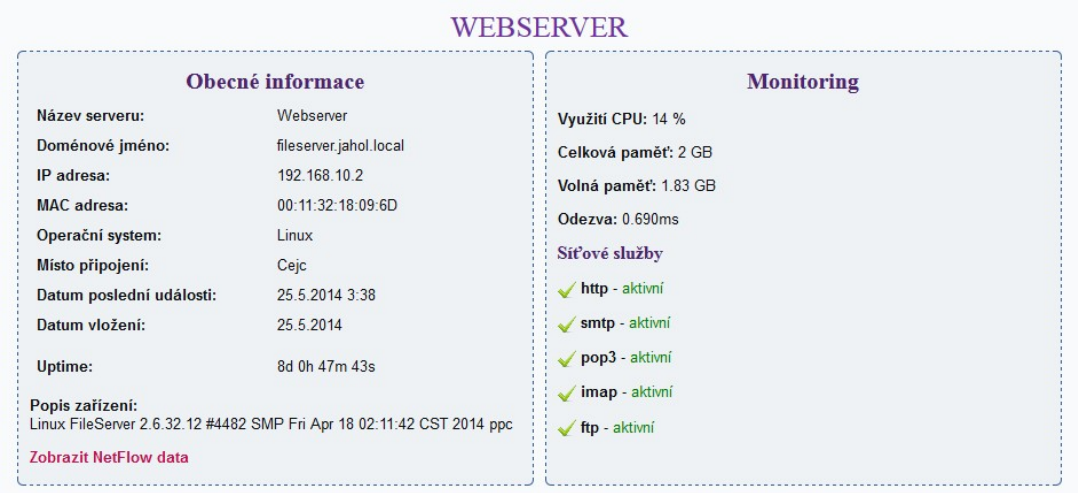
Prvním způsobem je právě využití vyhledaných zařízení (viz obrázek 5.4.1), kde zvolíme hledanou IP adresu, popř. adresu sítě s maskou sítě, a poté specifikujeme čísla otevřených portů. Jako výchozí adresa sítě je adresa získaná z fyzického síťového rozhraní hostovaného serveru systému AdminSys, takže uživatel nemusí zjišťovat používanou lokální adresu sítě. Čísla portů jsou rovněž ve výchozím nastavení předdefinována dle běžných serverových služeb, jako je doménová službu (port 53/TCP/UDP), poštovní služby (SMTP – 25/TCP, POP3 – 110/TCP) a další. Klikneme-li na tlačítko prohledat síť, budou nám vyhledány požadované zařízení přímo z databáze, které poté snadno uložíme do seznamu evidovaných serverů. V našem případě byla nalezena dostupná doménová služba, tj. port číslo 53.

Obrázek 5.4.1: Uložení nalezeného serveru

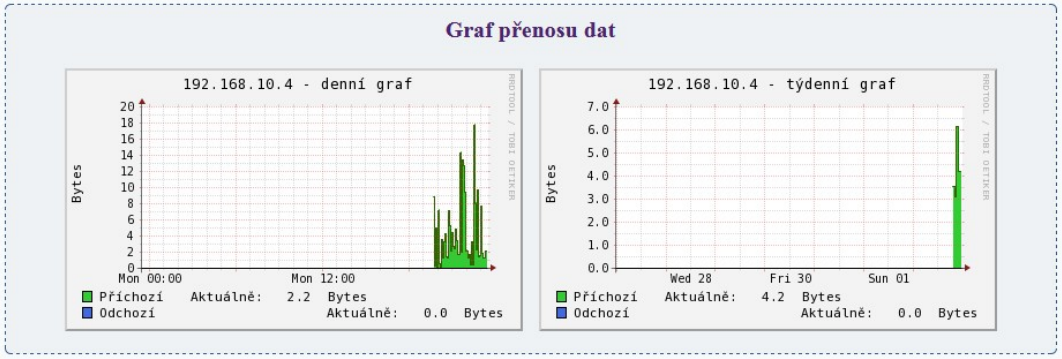
Druhým způsobem je podobně jako u klientských stanic ruční zadání alespoň konkrétní existující IP adresy zařízení a specifikování typu serveru z nabízeného seznamu. Zbylé informace o serveru, jako je název, doménové jméno a další, se systém pokusí opět sám dohledat. Podporuje-li zařízení SNMP protokol a při přidání tuto vlastnost také zvolíme, bude se systém primárně dotazovat na chybějící údaje přímo zařízení prostřednictvím SNMP. Dříve než tak učiníme, je potřeba zkontrolovat, zda je zvolena správná verze SNMP protokolu a název tzv. community řetězce. Ve výchozím nastavení má SNMP protokol hodnotu verze 1 a community řetězec jako public. Tento řetězec pak slouží jako heslo pro autentizaci a je uložen v textové podobě. Metoda, která zaopatřuje zjišťování informací pomocí SNMP protokolu se nazývá `getSnmpByAddr`. Pokud je funkce na straně zařízení nedostupná, systém zvolí alternativní název zařízení z DNS serveru, popř. použije IP adresu s prefixem `Host`.

```
public function getSnmpByAddr($address,$community,$version, $requestId) {
      $req = $this->snmp_req[$requestId];
      $result = "";
      if($req != NULL){
            snmp_set_quick_print(1);
            if($version == 1) {
                  $result = snmpget($address, $community, $req);
            } else {
                  $result = snmp2_get($address, $community, $req);
            }
      }
      return $result;
}
```

Pokud při přidávání nového serveru zvolíme správný typ zařízení, zpřístupní se nám aktivní monitorování předdefinovaných hodnot pomocí SNMP protokolu. Příklad takového serveru je znázorněn na obrázku 5.4.2. Nedílnou součástí je také sledování statistik přenosu dat podobě grafů (obrázek 5.4.3). Můžeme tak pasivně sledovat, který server byl v jakou dobu nejvíce datově zatížen.



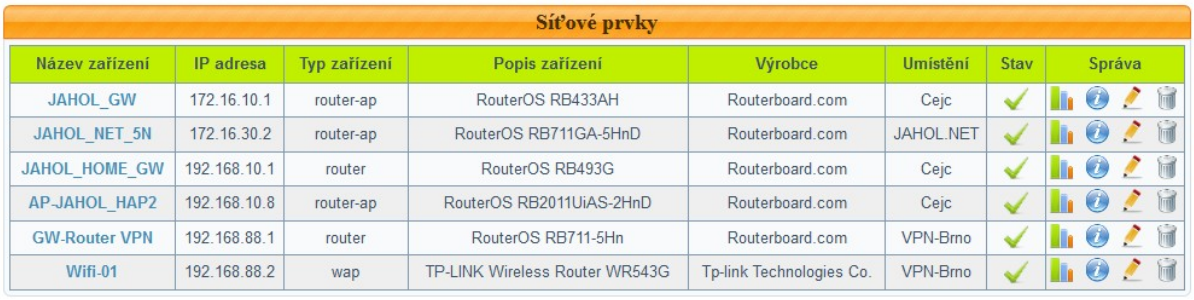
Obrázek 5.4.2: Detailní zobrazení serveru – monitorování



Obrázek 5.4.3: Detailní zobrazení serveru – graf přenosu dat

5.5 Monitorování síťových zařízení

Evidence a způsob aktivního a pasivního monitorování síťových prvků a síťových zařízení je obdobný, jako u serverových zařízení. Hlavní rozdíl je však v tom, že síťová zařízení fungují jako tzv. jednoúčelová zařízení s uzavřeným operačním systémem. Je zde tedy velmi malá pravděpodobnost selhání jednotlivých procesů. Proto jsem pro způsob monitorování těchto zařízení zvolil pouze metodu kontroly odezvy protokolem ICMP a v případě podpory ze strany zařízení také protokol SNMP. Přidání síťových zařízení do evidence systému je tak snadnější, můžeme-li využít dotazování na systémové informace přímo od zařízení. Postačí nám k tomu opět povolení SNMP a nastavení community řetězce včetně verze užívaného protokolu.

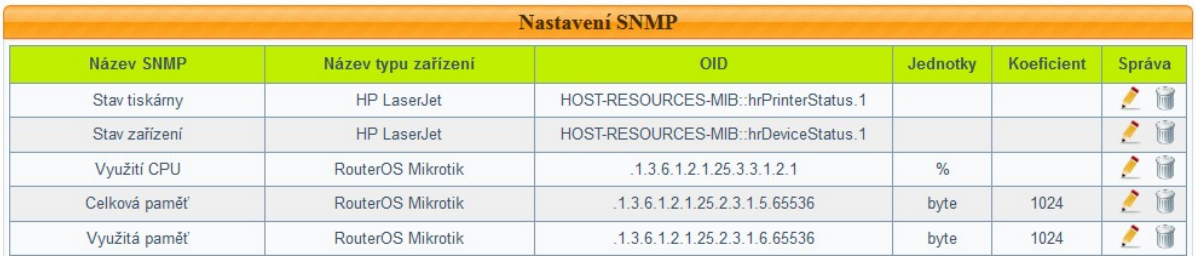


Síťové prvky

Název zařízení	IP adresa	Typ zařízení	Popis zařízení	Výrobce	Umístění	Stav	Správa
JAHOL_GW	172.16.10.1	router-ap	RouterOS RB433AH	Routerboard.com	Cejc	✓	
JAHOL_NET_5N	172.16.30.2	router-ap	RouterOS RB711GA-5HnD	Routerboard.com	JAHOL.NET	✓	
JAHOL_HOME_GW	192.168.10.1	router	RouterOS RB493G	Routerboard.com	Cejc	✓	
AP-JAHOL_HAP2	192.168.10.8	router-ap	RouterOS RB2011UiAS-2HnD	Routerboard.com	Cejc	✓	
GW-Router VPN	192.168.88.1	router	RouterOS RB711-5Hn	Routerboard.com	VPN-Brno	✓	
Wifi-01	192.168.88.2	wap	TP-LINK Wireless Router WR543G	Tp-link Technologies Co.	VPN-Brno	✓	

Obrázek 5.5.1: Přehled evidovaných síťových prvků

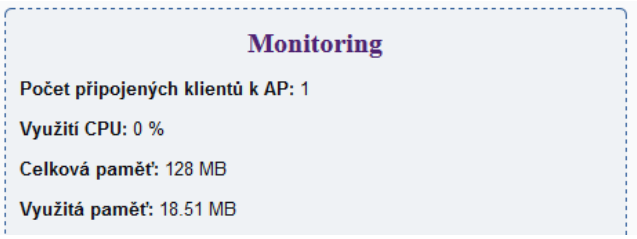
Díky tomu, že je testovaná počítačová síť z větší části homogenní (obrázek 5.5.1), je tak snadnější využití vytvořených profilů typů zařízení a k nim nadefinovaných monitorovacích SNMP identifikátorů (viz obrázek 5.5.2). Můžeme tak snadněji u všech zařízení stejné skupiny zobrazovat nově požadované údaje aktualizací jen jednoho profilu.



Nastavení SNMP

Název SNMP	Název typu zařízení	OID	Jednotky	Koeficient	Správa
Stav tiskárny	HP LaserJet	HOST-RESOURCES-MIB::hrPrinterStatus.1			
Stav zařízení	HP LaserJet	HOST-RESOURCES-MIB::hrDeviceStatus.1			
Využití CPU	RouterOS Mikrotik	.1.3.6.1.2.1.25.3.3.1.2.1	%		
Celková paměť	RouterOS Mikrotik	.1.3.6.1.2.1.25.2.3.1.5.65536	byte	1024	
Využitá paměť	RouterOS Mikrotik	.1.3.6.1.2.1.25.2.3.1.6.65536	byte	1024	

Obrázek 5.5.2: Seznam SNMP identifikátorů přiřazených k typům zařízení



Obrázek 5.5.3: Monitorování hodnot na základě profilů

5.6 Sledování síťového provozu

Pro sledování síťového provozu pomoci protokolu NetFlow jsem zvolil nejrozšířenější nástroj NFDump. Data jsou tak postupně zachytávána běžícím deamonem nfcapd a ukládána transparentně na disk. Systém AdminSys využívá pro ukládání přijatých dat úložiště na předem připraveném přídavném disku `/mnt/storage`.

```
# df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
                       17G  1.5G   15G  10% /
tmpfs                 250M     0  250M   0% /dev/shm
/dev/sda1             485M   50M  410M  11% /boot
/dev/sdb1             119G  270M  112G   1% /mnt/storage
```

Spuštění sběrače dat jako běžící službu provedeme následujícím způsobem:

```
# /usr/bin/nfcapd -D -l /mnt/storage/netflow/nfdump
```

Ověření běžící služby pak můžeme ověřit následovně:

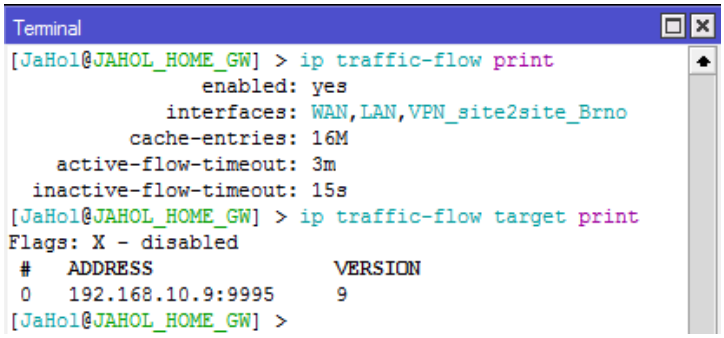
```
# lsof -Pni | grep nfcapd
nfcapd    5625    root    4u  IPv4  47668      0t0  UDP *:9995
```

Můžeme tedy vidět, že služba skutečně běží a naslouchá na UDP portu 9995. PID procesu je 5625. Je však ještě potřeba nastavit firewall pro zpřístupnění služby okolní síti. Můžeme zde specifikovat konkrétní zdrojové IP adresy zařízení, kterým povolím zasílání dat. Pro jednoduchost jsem povolil zasílání dat všem zařízením z lokální sítě.

```
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source       destination
ACCEPT     tcp  --  anywhere       anywhere    state NEW tcp dpt:https
ACCEPT     udp  --  anywhere       anywhere    state NEW udp dpt:palace-4
ACCEPT     all  --  anywhere       anywhere    state RELATED,ESTABLISHED
ACCEPT     icmp --  anywhere       anywhere
ACCEPT     all  --  anywhere       anywhere
ACCEPT     tcp  --  anywhere       anywhere    state NEW tcp dpt:ssh
ACCEPT     tcp  --  anywhere       anywhere    state NEW tcp dpt:http
REJECT     all  --  anywhere       anywhere    reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source       destination
REJECT     all  --  anywhere       anywhere     reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

```
# cat /etc/services | grep palace-4
palace-4        9995/tcp                # Palace-4
palace-4        9995/udp                # Palace-4
[root@AdminSys monitor]#
```

Následně jsem nastavil síťová zařízení pro zasílání dat směřovaná UDP port 9995 administrační server (viz obrázek 5.6.1).



Obrázek 5.6.1: Nastavení směrovače pro zasílání NetFlow dat

Nyní můžeme vyzkoušet pomocí nfdump zobrazit odchycená data:

```
# nfdump -R /mnt/storage/netflow/nfdump | head -5
Date first seen             Event   XEvent Proto        Src IP Addr:Port
Dst IP Addr:Port      X-Src IP Addr:Port          X-Dst IP Addr:Port    In
Byte Out Byte
2014-05-30 13:22:49.265 IGNORE   Ignore TCP         192.168.10.2:25     ->
195.154.10.37:49395             0.0.0.0:0       ->              0.0.0.0:0
839        0
2014-05-30 13:22:49.265 IGNORE   Ignore TCP        195.154.10.37:49395 ->
192.168.10.2:25                 0.0.0.0:0       ->              0.0.0.0:0
798        0
2014-05-30 13:22:52.635 IGNORE   Ignore UDP         192.168.10.2:58885 ->
192.168.10.1:53                 0.0.0.0:0       ->              0.0.0.0:0
144        0
```

Systém AdminSys nabízí snadnější vyhledávání záznamů prostřednicím uživatelsky přívětivějšího webového rozhraní (viz obrázek 5.6.2).

Výpis síťového provozu

Datum	Protokol	Zdrojová IP	Zdrojový port	Cílová IP	Cílový port	Počet bajtů	Počet paketů
2014-06-02 22:32:36.720	UDP	192.168.10.120	37231	192.168.10.1	53	128	2
2014-06-02 22:32:36.790	UDP	192.168.10.120	60229	192.168.10.1	53	128	2
2014-06-02 22:32:37.140	UDP	192.168.10.120	47840	192.168.10.4	53	136	2
2014-06-02 22:32:37.750	UDP	192.168.10.120	1126	192.168.10.4	53	124	2
2014-06-02 22:32:38.300	UDP	192.168.10.120	58102	192.168.10.4	53	148	2
2014-06-02 22:32:38.300	UDP	192.168.10.120	50317	192.168.10.4	53	126	2

Obrázek 5.6.2: Formulář s výsledkem nalezeného síťového provozu

5.7 Generování statistik

Statistiky v podobě grafů standardních síťových služeb a přenosu dat dle konkrétních IP adres nebo celkový přehled propustnosti sítě je generován pomocí RRDtool nástroje. Jak již bylo zmíněno v kapitole 4.2, pro každý graf je vytvořena jedna cyklická RRD databáze. Ta se v jazyce Perl vytvoří následujícím způsobem:

```
RRDs::create "database.rrd",
     "-s 300",
     "DS:svc:ABSOLUTE:600:U:U",
     "RRA:AVERAGE:0.5:1:576",
     "RRA:AVERAGE:0.5:6:672",
     "RRA:AVERAGE:0.5:24:732",
     "RRA:AVERAGE:0.5:144:1460";
```

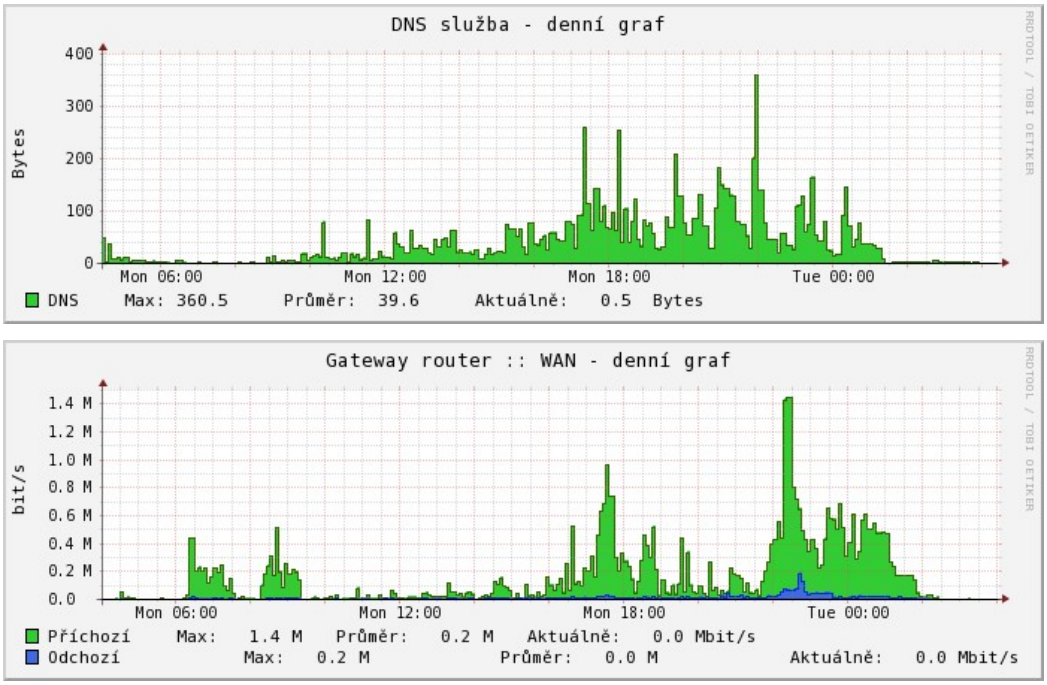
Vložení hodnoty do cyklické databáze provedeme příkazem `update` následovně:

```
RRDs::update "databáze.rrd","-t", "nazev_identifikatoru", "N:$data";
```

Vykreslení grafu z uložených statistik pak provedeme pomocí funkce `graph`:

```
RRDs::graph graf.png",
            "-s -1d",
            "-t Popis grafu",
            "--lazy",
            "-h", "140", "-w", "600",
            "-l 0",
            "-a", "PNG",
            "-v Jednotky",
            "DEF:svc=database.rrd:svc:AVERAGE",
            "AREA:svc#32CD32:$_[0]",
            "LINE1:svc#336600",
            "GPRINT:svc:MAX:  Max\\: %5.1lf %s",
            "GPRINT:svc:AVERAGE: Průměr\\: %5.1lf %S",
            "GPRINT:svc:LAST: Aktuálně\\: %5.1lf %SJednotky\\n",
            "HRULE:0#000000";
```

Výsledný graf síťových služeb nebo celkového přenosu síťových dat poté vypadá jako obrázky 5.7.



Obrázek 5.7: Grafické statistiky síťového provozu

6 Instalace

Přiložený DVD disk obsahuje jak samostatné zdrojové kódy, tak virtualizovaný server v podobě image. Přihlašovací login pro správu serveru je root a heslo @dm1n$ys. Pro přihlášení do webové aplikace je uživatelské jméno spravce a heslo adminsys. V případě, že bude AdminSys provozován na jiném serveru, je potřeba doinstalovat balíčky pro podporu NetFlow, Nmap a RRDtool. Očekává se také přítomnost standardně dostupného jazyka Perl. Přiložen je také soubor s databázovými SQL příkazy včetně vzorových dat. Skriptem install.sh se provede instalace deamona pro sběr a zpracování síťových dat. Zajistí se také automatické spuštění po restartu serveru.

7 Závěr

Cílem bylo navrhnout a implementovat co nejvíce automatizovaný systém pro správu počítačové sítě. Využitím vhodně zvolených nástrojů toto bylo dodrženo. Systém byl rovněž testován v produkční síti (viz obrázek 3.1). Ačkoliv byla původně zamýšlena implementace systému s mnohem více funkcemi, bylo toto z časového důvodu přehodnoceno a ponecháno jako další možná rozšíření. Jedná se konkrétně o zpracování systémových dat Syslog a upozorňování správce sítě na události při výpadku zařízení. Systém je tak ve fázi dohledového centra, který pouze informuje o stavu zařízení prohlížením evidovaných zařízení. Při implementaci jsem se potýkal s problémy Nette frameworku, který byl značně inovován, a původní dokumentace pro starší verzi, kterou jsem zpočátku využíval, byla přepsána.

Literatura

[1] KABELOVÁ, Alena a Libor DOSTÁLEK. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.

[2] Hlavní přednosti Nette Frameworku. *Nette Framework* [online]. 2014 [cit. 2014-01-12]. Dostupné z: http://nette.org/cs/#toc-features

[3] Princip MVP (a MVC). *Nette Framework* [online]. 2010 [cit. 2014-01-12]. Dostupné z: http://doc.nette.org/cs/0.9/quickstart/vytvoreni-presenteru

[4] DOČEKAL, Michal. Správa linuxového serveru: Skenování sítí pomocí Nmap. *LinuxExpres* [online]. 2012 [cit. 2014-01-12]. Dostupné z: http://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-typy-skenu-a-volby-nmapu

[5] Service and Version Detection. *Nmap Reference Guide* [online]. 2011 [cit. 2014-01-12]. Dostupné z: http://nmap.org/book/man-version-detection.html

[6] BLANK-EDELMAN, David N a David N BLANK-EDELMAN. *Automating system administration with Perl*. 2nd ed., Rev. Cambridge: O'Reilly, c2009, xxiii, 639 p. ISBN 05-960-0639-X.

[7] Perl for System Administration: Security and Network Monitoring. *Perl for System Administration* [online]. 2001 [cit. 2014-01-12]. Dostupné z: http://docstore.mik.ua/orelly/perl/sysadmin/ch10_03.htm

[8] OETIKER, Tobias. RRDtool. *RRDtool Tutorial* [online]. 2009 [cit. 2014-01-12]. Dostupné z: http://oss.oetiker.ch/rrdtool/tut/rrd-beginners.en.html

[9] Quick Start - dibi. *Http://dibiphp.com/cs/quick-start* [online]. 2008 [cit. 2014-01-12]. Dostupné z: http://dibiphp.com/cs/quick-start

[10] NFDUMP. *Http://nfdump.sourceforge.net/* [online]. 2013 [cit. 2014-01-12]. Dostupné z: http://nfdump.sourceforge.net/

[11] VirtualBoxes: Free VirtualBox® Images. *Http://virtualboxes.org/images/centos/* [online]. 2013 [cit. 2014-01-12]. Dostupné z: http://virtualboxes.org/images/centos/